Improved Conditioning of Finite Element Matrices using New High Order Interpolatory Bases

R. Rieben D. White and G. Rodrigue

***Abstract*—**

The condition number of finite element matrices constructed from interpolatory bases will grow as the polynomial degree of the basis functions is increased. The worst case scenario for this growth rate is exponential and in this paper we demonstrate through computational example that the traditional set of uniformly distributed interpolation points yields this behavior. We propose a set of non-uniform interpolation points which yield a much improved polynomial growth rate of condition number. These points can be used to construct several types of popular hexahedral basis functions including the 0-form (standard Lagrangian), 1-form (Curl conforming) and 2-form (Divergence conforming) varieties. We demonstrate through computational example the benefits of using these new interpolatory bases in finite element solutions to Maxwell's equations in both the frequency and time domain.

I. Introduction

We are concerned with the finite element solution of Maxwell's equations on unstructured hexahedral grids. This process generally involves the solution of large linear systems and for the case of time domain simulations, these systems may need to be solved one or more times per simulation time step. As such the conditioning of these linear systems becomes increasingly important as their sizes increase [1], [2]. Recently, high order finite element methods have been developed and used with great success in the field of computational electromagnetics (CEM) [3]. High-order methods can yield extremely accurate and efficient results for certain problems with smoothly curved boundaries, and they can drastically reduce the effects of numerical dispersion [4], [5]. Such methods are based on a set of polynomial basis functions which are in turn built from a set of interpolatory polynomials. As the polynomial degree p is increased, the eigenvalue spectrum of the resulting finite element matrices will grow, thus causing the condition number of the matrix to grow as well. The worst case scenario is exponential growth of matrix condition number as a function of p.

Several investigations into the conditioning of finite element matrices have been performed using various types of bases. In [6], the conditioning of finite element solutions to the two dimensional scalar elliptic problem is investigated using a class of interpolatory basis functions which use the Gauss-Lobatto interpolation points. It is shown that a condition number growth rate of $O(p)$ can be achieved for the stiffness matrix. In [7], the condition number of diagonally scaled matrices is investigated using hierarchic Nedelec curl-conforming (or 1-form) basis functions on hexahedral elements. It is shown that an upper bound for the growth of the condition number of the global stiffness matrix using these hierarchical bases is $O(p^4)$ for problems in which the characteristic frequency is much greater than p. In [8], a set of hierarchical 1-form basis functions are defined for the frequency domain vector Helmholtz equation. The basis functions are designed to be nearly orthogonal over tetrahedral meshes, yielding rapid convergence for problems with regular domains. In [9], it is shown that the preconditioned conjugate gradient (PCG) method is scalable when used to invert mass matrices arising from vector finite element discretizations of the time domain Maxwell equations. For the case of first order basis functions, it is also shown that the condition number of the diagonally scaled mass matrix remains constant as the size of the mesh increases provided the ratio of the mesh lengths remains constant.

In this paper we demonstrate a way to achieve polynomial growth as a function of p for the special case of hexahedral interpolatory bases. This growth rate is achieved by using a special set of non-uniformly spaced interpolation points in conjunction with a new set of polynomial basis functions (and subsequent degrees of freedom). We demonstrate the improved iterative solution time obtained with these new basis functions for a series of frequency and time domain problems using a variety of iterative solvers and preconditioners.

In this paper we use differential forms as a convenient way of classifying the various field types involved in solving Maxwell's equations. In addition, the calculus of differential forms provides the necessary transformation rules which allow complicated basis functions to be derived on a reference element and then mapped to global mesh elements. Table I lists various physical quantities in electromagnetics and their associated differential form.

II. Choice of Interpolation Points

The Lagrange interpolatory polynomial of degree p is defined by a distinct set of $p+1$ real valued interpolation points denoted by the symbol X, such that $X = \{X_0, X_1, \ldots, X_p\}$. The polynomial is constructed in such a way that it has a value of unity at interpolation point i and a value of zero at every other

This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48, and under the U.S. Air Force Contract No. F49620-01-1-0327

University of California, Davis and Institute for Scientific Computing Research, Lawrence Livermore National Laboratory, rieben1@llnl.gov

Defense Sciences Engineering Division, Lawrence Livermore National Laboratory, white37@llnl.gov

University of California, Davis and Institute for Scientific Computing Research, Lawrence Livermore National Laboratory, rodrigue@ucdavis.edu

Physical Quantity	Units	Differential Form
Scalar Potential	V/m^0	0-form
Electric Field Intensity	V/m^1	1-form
Magnetic Field Intensity	A/m^1	1-form
Electric Flux Density	C/m^2	2-form
Magnetic Flux Density	W/m^2	2-form
Electric Charge Density	C/m^3	3-form

TABLE I

PHYSICAL QUANTITIES AND THEIR ASSOCIATED DIFFERENTIAL FORMS

interpolation point. The precise definition for the Lagrange interpolatory polynomial of degree p is given by

$$L_i^p(x;X) = \prod_{\substack{j=0 \\ j\neq i}}^{p} \frac{(x-X_j)}{(X_i-X_j)} \tag{1}$$

The set of $p+1$ interpolation points, X, can at this point be arbitrary. Traditionally, the set X consists of $p+1$ interpolation points uniformly distributed over the unit interval. This set is simple to generate but turns out to yield the worst case scenario of exponential growth of condition number as a function of p, as shown in Sections V – VII.

In the field of approximation theory, various sets of interpolation points over the unit interval have been investigated. For a given value of p, there exists an optimal set of interpolation points which will yield a minimum Lebesgue constant (a measure of the interpolating polynomial's performance); unfortunately there is no analytic formula for generating these points in an efficient manner. However, it turns out that the zeros of certain orthogonal polynomials can yield near optimal results. For example, Chebyshev points or Gauss-Lobatto points can be used to construct nearly optimal interpolatory polynomials. We follow the results of [10] and note that there is a near optimal set of points that can be computed quite easily. This set of points, referred to as the extended Chebyshev set, is generated by computing the zeros of the Chebyshev polynomial of the first kind, then applying a linear transformation to map the results over the domain $[0,1]$. Let $\bar{X}^p$ denote the extended Chebyshev set of $p+1$ interpolation points over the domain $[0,1]$ defined as

$$\bar{X}^p = \{\frac{-\cos[(2i+1)\pi/(2p+2)]}{2\cos[\pi/(2p+2)]} + \frac{1}{2};\ \ i=0,1,\ldots,p\} \tag{2}$$

While optimal interpolation points X have been investigated from the point of view of approximation, we show via computational experiments that the conditioning of finite element matrices is strongly dependent upon the choice of interpolation points, with uniform points yielding exponential growth of condition number, and non-uniform points yielding polynomial growth of condition number.

III. BASIS FUNCTIONS

We now present explicit formulae for the construction of interpolatory basis functions. We perform all computations on a reference element $\hat{\Omega}$ (all objects explicitly defined on the reference element will be accented with a *hat* symbol). There exists a mapping Φ from the reference element $\hat{\Omega}$ to the actual element Ω. This mapping (defined by interpolatory shape functions) and its Jacobian are defined as

$$\mathbf{r} = \Phi(\hat{\mathbf{r}});\quad \mathbf{J}_{i,j} = \frac{\partial r_j}{\partial \hat{r}_i} \tag{3}$$

where $\hat{\mathbf{r}} \in \hat{\Omega}$ and $\mathbf{r} \in \Omega$. Unlike the approach presented in [3], we define all basis functions on the reference element and transform them as necessary during the finite element assembly procedure. As such, we also present the appropriate transformation rules which map the basis functions from the reference element to physical mesh elements.

For clarity, we will denote the three independent variables in the reference system using the standard Cartesian notation of $(\hat{x},\hat{y},\hat{z})$. In addition, we will denote contra-variant basis vectors as $\hat{\mathbf{x}},\hat{\mathbf{y}}$ and $\hat{\mathbf{z}}$, while covariant basis vectors will be denoted as $\hat{\mathbf{X}},\hat{\mathbf{Y}}$ and $\hat{\mathbf{Z}}$. This is a trivial distinction since these basis vectors are identical, have unit magnitude, and are constant over the domain of the reference hexahedron. However, we make the distinction to emphasize the different transformation properties of the bases associated with these vectors. We also omit the interpolation points X from the Lagrange interpolatory polynomials, implying that the set may be arbitrary; however, for improved matrix conditioning the extended Chebyshev points from (2) should be used. In order to ensure the proper conformity across element to element interfaces, it is crucial that the basis functions (and consequently the degrees of freedom) be associated with the various sub-simplices of the element (e.g. nodes, edges, faces, etc ...). This property is referred to as *locality* [11]. As such, we decompose all of the bases into subsets corresponding to the sub-simplex they are associated with. The dimensions of these subsets will in general be *number of sub-simplices per element* times *number of degrees of freedom per sub-simplex*.

A. 0-form Basis Functions

Let $\hat{N}$ denote a 0-form basis on the reference element. In order to satisfy the locality property, we can break this set of basis functions into four mutually disjoint subsets such that

$$\hat{N} = \hat{N}_n \cup \hat{N}_e \cup \hat{N}_f \cup \hat{N}_v \tag{4}$$

where the subscripts n,e,f and v denote the nodes, edges, faces and volume of the reference element respectively. For 0-forms, locality implies that the node basis functions should have non-zero values at one and only one node. The edge basis functions will have a non-zero value along one and only one edge and zero values at all of the nodes. The face basis functions will have non-zero values inside one and only one face with zero values at all of the nodes and along all of the edges. Finally, the volume basis functions will be zero at all of the nodes, along all of the edges and inside all of the faces. The 0-form node basis functions of polynomial degree p are given by

$$\hat{N}_n = \left\{L_i^p(\hat{x})L_j^p(\hat{y})L_k^p(\hat{z});\quad i,j,k=0,p\right\} \tag{5}$$

yielding 1 basis function per node. The 0-form edge basis functions of polynomial degree p are given by

$$\hat{N}_e = \begin{cases} L_i^p(\hat{y})L_j^p(\hat{z})L_k^p(\hat{x}) \\ L_i^p(\hat{x})L_j^p(\hat{z})L_k^p(\hat{y}) \\ L_i^p(\hat{x})L_j^p(\hat{y})L_k^p(\hat{z}) \end{cases} \tag{6}$$

$$i,j = 0,p; \;\; k = 1,\dots,p-1$$

The indices i and j loop over the 4 edges parallel to either the x,y or z axis. The index k loops over the $p-1$ basis functions per edge for a total of $12(p-1)$. The 0-form face basis functions of polynomial degree p are given by

$$\hat{N}_f = \begin{cases} L_i^p(\hat{x})L_j^p(\hat{y})L_k^p(\hat{z}) \\ L_i^p(\hat{y})L_j^p(\hat{x})L_k^p(\hat{z}) \\ L_i^p(\hat{z})L_j^p(\hat{x})L_k^p(\hat{y}) \end{cases} \tag{7}$$

$$i = 0,p; \;\; j,k = 1,\dots,p-1$$

The index i loops over the 2 faces that are normal to either the x,y, or z axis. The indices j and k loop over the $(p-1)^2$ basis functions in each face for a total of $6(p-1)^2$. Finally, there will be a total of $(p-1)^3$ 0-form basis functions that are internal to the reference element (i.e. functions not shared between elements), given by

$$\hat{N}_v = \begin{cases} L_i^p(\hat{x})L_j^p(\hat{y})L_k^p(\hat{z}) \\ L_i^p(\hat{y})L_j^p(\hat{z})L_k^p(\hat{x}) \\ L_i^p(\hat{z})L_j^p(\hat{x})L_k^p(\hat{y}) \end{cases} \tag{8}$$

$$i,j,k = 1,\dots,p-1$$

B. 1-form Basis Functions

Let $\hat{W}$ denote a 1-form basis on the reference element. In order to satisfy the locality property, we can break this set of basis functions into three mutually disjoint subsets such that

$$\hat{W} = \hat{W}_e \cup \hat{W}_f \cup \hat{W}_v \tag{9}$$

where the subscripts e, f and v denote the edges, faces and volume of the reference element respectively. For 1-forms, locality implies that the edge basis functions should have non-vanishing tangential components along one and only one edge. The face basis functions will have non-vanishing tangential components along one and only one face with no tangential components along any edges. Finally, the volume basis functions will have no tangential components along either edges or faces. The 1-form edge basis functions of polynomial degree p are given by

$$\hat{W}_e = \begin{cases} L_i^p(\hat{y})L_j^p(\hat{z})L_k^{p-1}(\hat{x})\;\hat{\mathbf{x}} \\ L_i^p(\hat{x})L_j^p(\hat{z})L_k^{p-1}(\hat{y})\;\hat{\mathbf{y}} \\ L_i^p(\hat{x})L_j^p(\hat{y})L_k^{p-1}(\hat{z})\;\hat{\mathbf{z}} \end{cases} \tag{10}$$

$$i,j = 0,p; \;\; k = 0,\dots,p-1$$

This set of functions is grouped into three sub-sets, one for each contravariant basis vector. The indices i and j loop over the 4 edges that are tangent to these basis vectors. The index k loops over the p basis functions per edge for a total of $12p$. The 1-form face basis functions of polynomial degree p are given by

$$\hat{W}_f = \begin{cases} L_i^p(\hat{x})L_j^p(\hat{z})L_k^{p-1}(\hat{y})\;\hat{\mathbf{y}} \\ L_i^p(\hat{x})L_j^p(\hat{y})L_k^{p-1}(\hat{z})\;\hat{\mathbf{z}} \\ L_i^p(\hat{y})L_j^p(\hat{z})L_k^{p-1}(\hat{x})\;\hat{\mathbf{x}} \\ L_i^p(\hat{y})L_j^p(\hat{x})L_k^{p-1}(\hat{z})\;\hat{\mathbf{z}} \\ L_i^p(\hat{z})L_j^p(\hat{y})L_k^{p-1}(\hat{x})\;\hat{\mathbf{x}} \\ L_i^p(\hat{z})L_j^p(\hat{x})L_k^{p-1}(\hat{y})\;\hat{\mathbf{y}} \end{cases} \tag{11}$$

$$i = 0,p; \;\; j = 1,\dots,p-1; \;\; k = 0,\dots,p-1$$

This set of functions is grouped into six sub-sets, two for each face representing the contravariant basis vectors that are in the plane of that face. The index i loops over the 2 faces that are coplanar to these basis vectors. The indices j and k loop over the $2p(p-1)$ basis functions per face for a total of $12p(p-1)$. Finally, there will be a total of $3p(p-1)^2$ interpolatory basis functions that are internal to the reference element given by

$$\hat{W}_v = \begin{cases} L_i^p(\hat{y})L_j^p(\hat{z})L_k^{p-1}(\hat{x})\;\hat{\mathbf{x}} \\ L_i^p(\hat{x})L_j^p(\hat{z})L_k^{p-1}(\hat{y})\;\hat{\mathbf{y}} \\ L_i^p(\hat{x})L_j^p(\hat{y})L_k^{p-1}(\hat{z})\;\hat{\mathbf{z}} \end{cases} \tag{12}$$

$$i,j = 1,\dots,p-1; \;\; k = 0,\dots,p-1$$

C. 2-form Basis Functions

Let $\hat{F}$ denote a 2-form basis on the reference element. In order to satisfy the locality property, we can break this set of basis functions into two mutually disjoint subsets such that

$$\hat{F} = \hat{F}_f \cup \hat{F}_v \tag{13}$$

where the subscripts f and v denote the faces and volume of the reference element respectively. For 2-forms, locality implies that the face basis functions will have non-vanishing normal components along one and only one face while the volume basis functions will have no normal components along any of the the faces. The interpolatory face basis functions of polynomial degree p are given by

$$\hat{F}_f = \begin{cases} L_i^p(\hat{x})L_j^{p-1}(\hat{y})L_k^{p-1}(\hat{z})\;\hat{\mathbf{X}} \\ L_i^p(\hat{y})L_j^{p-1}(\hat{x})L_k^{p-1}(\hat{z})\;\hat{\mathbf{Y}} \\ L_i^p(\hat{z})L_j^{p-1}(\hat{x})L_k^{p-1}(\hat{y})\;\hat{\mathbf{Z}} \end{cases} \tag{14}$$

$$i = 0,p; \;\; j,k = 0,\dots,p-1$$

This set of functions is grouped into three sub-sets, one for each of the covariant basis vectors. The index i loops over the 2 faces that are normal to these basis vectors. The indices j and k loop over the p^2 basis functions per face for a total of $6p^2$. Finally, there will be a total of $3p^2(p-1)$ interpolatory basis functions that are internal to the reference element given by

$$\hat{F}_v = \begin{cases} L_i^p(\hat{x})L_j^{p-1}(\hat{y})L_k^{p-1}(\hat{z})\;\hat{\mathbf{X}} \\ L_i^p(\hat{y})L_j^{p-1}(\hat{x})L_k^{p-1}(\hat{z})\;\hat{\mathbf{Y}} \\ L_i^p(\hat{z})L_j^{p-1}(\hat{x})L_k^{p-1}(\hat{y})\;\hat{\mathbf{Z}} \end{cases} \tag{15}$$

$$i = 1,\dots,p-1; \;\; j,k = 0,\dots,p-1$$

D. Basis Function Transformation Rules

The basis functions presented are valid only in the reference system. We therefore need a general procedure for transforming functions defined in one coordinate system to another system. In order to maintain coordinate independence, all properties of the functions defined in one coordinate system must be preserved under a transformation to a new coordinate system, this property is known as *invariance* [11]. For example, invariance implies that 1-form basis functions defined to have non-vanishing tangential components along only one edge in the reference coordinate system, must also have non-vanishing tangential components along only the same edge in the new coordinate system. In addition, we would like the scaling of the functions to be independent of the coordinate system used to represent them. Table II gives the precise transformation rules for 0-forms, 1-forms and 2-forms, their respective exterior derivatives and the units of these transformations; all of which can be derived using the calculus of differential forms. The symbol m denotes an arbitrary metric of distance while the symbol $\circ$ denotes composition.

Object	**Transformation Rule**	**Units**
0-form functions	$n \circ \Phi = \hat{n}$	m^{-0}
Grad of 0-form	$(\nabla n) \circ \Phi = \mathbf{J}^{-1}(\nabla \hat{n})$	m^{-1}
1-form functions	$\mathbf{w} \circ \Phi = \mathbf{J}^{-1}\hat{\mathbf{w}}$	m^{-1}
Curl of 1-form	$(\nabla \times \mathbf{w}) \circ \Phi = \frac{1}{\lvert\mathbf{J}\rvert}\mathbf{J}^T(\nabla \times \hat{\mathbf{w}})$	m^{-2}
2-form functions	$\mathbf{f} \circ \Phi = \frac{1}{\lvert\mathbf{J}\rvert}\mathbf{J}^T\hat{\mathbf{f}}$	m^{-2}
Div of 2-form	$(\nabla \cdot \mathbf{f}) \circ \Phi = \frac{1}{\lvert\mathbf{J}\rvert}(\nabla \cdot \hat{\mathbf{f}})$	m^{-3}

TABLE II
TRANSFORMATION RULES

IV. Degrees of Freedom

In [12], a set of integral based degrees of freedom (DOF) are presented which are presumed to be computed exactly. In practice, such DOF can be computationally expensive to implement and cannot always be integrated exactly using numerical quadrature. As such, we present a set of discrete DOF that are based on evaluation of a function at a point. These point based DOF satisfy the properties of invariance and locality and given a set of basis functions, they can be used to enforce unisolvence. Explicit knowledge of the DOF are not required for formation of the mass and stiffness matrices, and hence are not presented in many publications. However, the DOF are required for an explicit interpolation operation which is used for many things including the implementation of arbitrary boundary conditions, projection of a solution onto a finer mesh (mesh refinement) or on to a coarser mesh (multi-grid algorithms), and for special mixed bilinear forms. For example, the 1-form interpolation operator looks like

$$\mathbf{g} \approx \Pi(\mathbf{g}) = \sum_{i=1}^{dim(W)} \mathcal{A}_i(\mathbf{g})\,\mathbf{w}_i; \quad \mathbf{w}_i \in W \tag{16}$$

where the interpolation operator $\Pi(\mathbf{g})$ denotes a basis function expansion of $\mathbf{g}$. It is also important to point out that unlike the DOF for the bases of [3], our DOF are scale independent. Because the basis functions of [3] are fully interpolatory, they have no physical units associated with them; instead the DOF carry the physical units of whatever field the basis functions are discretizing, e.g. the curl-conforming basis functions are dimensionless while the corresponding degrees of freedom have units of V/m for the case of the electric field. Because our interpolatory basis functions are based on differential forms, they have purely spatial units of m^{-l}, where l is the degree of the form. The following degrees of freedom will always carry the physical units of the field that are not related to space (such as voltage, current, charge, etc ...) and will always be scale invariant.

Let X denote a set of $p+1$ interpolation points over the unit interval $[0,1]$ and X' denote a set of p interpolation points over the same interval. The 0-form point degrees of freedom are given by

$$\mathcal{A}(n) = \left\{ n(\Phi(X_i, X_j, X_k)); \quad i,j,k = 0,\ldots,p \right\} \tag{17}$$

The 1-form point degrees of freedom are given by

$$\mathcal{A}(\mathbf{w}) = \begin{cases} \mathbf{w}(\Phi(X'_i, X_j, X_k)) \cdot \mathbf{J}^T\hat{\mathbf{x}} \\ \mathbf{w}(\Phi(X_k, X'_i, X_j)) \cdot \mathbf{J}^T\hat{\mathbf{y}} \\ \mathbf{w}(\Phi(X_j, X_k, X'_i)) \cdot \mathbf{J}^T\hat{\mathbf{z}} \end{cases} \tag{18}$$

$$i = 0,\ldots,p-1; \quad j,k = 0,\ldots,p$$

where $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ and $\hat{\mathbf{z}}$ denote the contravariant basis vectors on the reference element. The 2-form point degrees of freedom are given by

$$\mathcal{A}(\mathbf{f}) = \begin{cases} \mathbf{f}(\Phi(X_i, X'_j, X'_k)) \cdot |\mathbf{J}|\mathbf{J}^{-1}\hat{\mathbf{X}} \\ \mathbf{f}(\Phi(X'_k, X_i, X'_j)) \cdot |\mathbf{J}|\mathbf{J}^{-1}\hat{\mathbf{Y}} \\ \mathbf{f}(\Phi(X'_j, X'_k, X_i)) \cdot |\mathbf{J}|\mathbf{J}^{-1}\hat{\mathbf{Z}} \end{cases} \tag{19}$$

$$i = 0,\ldots,p; \quad j,k = 0,\ldots,p-1$$

where $\hat{\mathbf{X}}, \hat{\mathbf{Y}}$ and $\hat{\mathbf{Z}}$ denote the covariant basis vectors on the reference element.

The unisolvence property for degrees of freedom requires that

$$\mathcal{A}_i(g_j) = \delta_{i,j} \tag{20}$$

for basis functions g_j. This property must hold in order for basis function expansions to be valid. In order to enforce unisolvence for a given basis, we first construct the matrix

$$V_{i,j} = \mathcal{A}_i(g_j) \tag{21}$$

This matrix forms a linear mapping that is similar to a Vandermonde matrix. We can now apply this linear mapping to either the basis g_j or the degrees of freedom $\mathcal{A}$ in order to enforce unisolvence. In this paper we have presented basis functions with particular properties that we would like to preserve; as such we will apply the linear mapping to $\mathcal{A}$ in order to satisfy unisolvence. Because the degrees of freedom are linear functionals, we can construct a new set of degrees of freedom, denoted $\mathcal{A}'$, by the relation

$$\mathcal{A}' = (V^{-1})^T \mathcal{A} \tag{22}$$

V. Bilinear Forms

In the Galerkin finite element procedure, we require bilinear forms to construct a system of linear equations. Typically, these bilinear forms involve integrals of basis functions, their derivatives and possibly material property functions (i.e. dielectric or magnetic property functions) over the volume of mesh elements. They are used to construct objects such as global mass and stiffness matrices. In general, elements from an unstructured mesh will have non-trivial geometries, and as such, integration over the actual elements can be cumbersome and computationally expensive. However, integration over the standard reference element can be done quite easily and since the bases are polynomial in nature, integration can quite often be done exactly using a quadrature rule of the appropriate order. In addition, given the appropriate transformation rules the bases need only be evaluated on the reference element then transformed accordingly. This gives rise to a very computationally efficient algorithm for computing finite element approximations. For a given element topology and basis order,the basis functions only need to be computed and sampled at the quadrature points once. Then, for every element of the same topology in the mesh, the results from the reference element can simply be mapped according to the transformation rules. This can significantly reduce computational time and storage requirements for a typical finite element computation

In the following explicit bilinear forms, M_α denotes a mass matrix with a material property function α defined over an element Ω while S_γ denotes a stiffness matrix with a material property function γ defined over an element Ω. The material property functions are free to be (possibly tensor valued) functions of space and will affect the scaling of each bilinear form. For 0-forms, we have the following symmetric bilinear forms

$$M_\alpha\langle n_i, n_j\rangle = \int_{\hat{\Omega}} ((\alpha \circ \Phi)\hat{n}_i)\ \hat{n}_j\ |\mathbf{J}| \tag{23}$$

$$S_\gamma\langle n_i, n_j\rangle = \int_{\hat{\Omega}} ((\gamma \circ \Phi)\mathbf{J}^{-1}\nabla\hat{n}_i) \cdot (\mathbf{J}^{-1}\nabla\hat{n}_j)\ |\mathbf{J}| \tag{24}$$

The 0-form mass matrix will have units of (or scale as) $\alpha\, m^3$ while the 0-form stiffness matrix will scale as $\gamma\, m^1$, where m denotes an arbitrary metric of distance. For 1-forms, we have the following symmetric bilinear forms

$$M_\alpha\langle \mathbf{w}_i, \mathbf{w}_j\rangle = \int_{\hat{\Omega}} ((\alpha \circ \Phi)\mathbf{J}^{-1}\hat{\mathbf{w}}_i) \cdot (\mathbf{J}^{-1}\hat{\mathbf{w}}_j)\ |\mathbf{J}| \tag{25}$$

$$S_\gamma\langle \mathbf{w}_i, \mathbf{w}_j\rangle = \int_{\hat{\Omega}} ((\gamma \circ \Phi)\frac{1}{|\mathbf{J}|}\mathbf{J}^T(\nabla \times \hat{\mathbf{w}}_i)) \cdot (\frac{1}{|\mathbf{J}|}\mathbf{J}^T(\nabla \times \hat{\mathbf{w}}_{\mathbf{j}}))\ |\mathbf{J}| \tag{26}$$

The 1-form mass matrix will scale as $\alpha\, m^1$ while the 1-form stiffness matrix will scale as $\gamma\, m^{-1}$. Finally for 2-forms, we have the following symmetric bilinear forms

$$M_\alpha\langle \mathbf{f}_i, \mathbf{f}_j\rangle = \int_{\hat{\Omega}} ((\alpha \circ \Phi)(\frac{1}{|\mathbf{J}|}\mathbf{J}^T\hat{\mathbf{f}}_i)) \cdot (\frac{1}{|\mathbf{J}|}\mathbf{J}^T\hat{\mathbf{f}}_j)\ |\mathbf{J}| \tag{27}$$

$$S_\gamma\langle \mathbf{f}_i, \mathbf{f}_j\rangle = \int_{\hat{\Omega}} ((\gamma \circ \Phi)\frac{1}{|\mathbf{J}|}(\nabla \cdot \hat{\mathbf{f}}_i))\ (\frac{1}{|\mathbf{J}|}(\nabla \cdot \hat{\mathbf{f}}_{\mathbf{j}}))\ |\mathbf{J}| \tag{28}$$

The 2-form mass matrix will scale as $\alpha\, m^{-1}$ while the 2-form stiffness matrix will scale as $\gamma\, m^{-3}$.

Given the previously defined bilinear forms, it is now possible to construct local (or single element) matrices and investigate their conditioning as a function of polynomial degree p. Figures 1 - 3 show plots of the base 10 log of the condition number of single element mass matrices constructed using the explicit bilinear forms of (23), (25) and (27) in conjunction with the basis functions from section III using two different sets of interpolation points: uniform points and the extended Chebyshev points of (2). For the vector valued 1-form and 2-form basis functions, we use the shifted uniform interpolation scheme of [3]. Note how the uniformly spaced interpolation points yield exponential growth of condition number while the extended Chebyshev points yield a much improved polynomial growth.

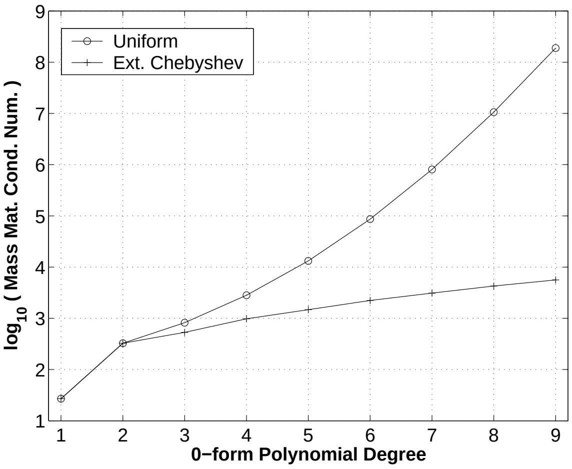


Fig. 1. Condition number of 0-form mass matrix using two different sets of interpolation points.

VI. Frequency Domain Examples

We now present some numerical results in the frequency domain demonstrating the improved performance of the newly proposed interpolatory bases. These examples involve the solution of three different PDEs using both uniform and extended Chebyshev interpolation points. In each example, the problem is discretized over a hexahedral mesh of a unit cube subject to a Dirichlet boundary condition and the resulting linear systems are solved iteratively to a prescribed error tolerance. Each of the PDEs can be represented discretely as a linear system of the form

$$\left(S_\gamma + (-1)^l\ \omega^2\ M_\alpha\right) g = (-1)^{l+1}\, s \tag{29}$$

where S_γ and M_α are stiffness and mass matrices respectively computed using the appropriate bilinear form of section V, g is the discrete vector representation of the field unknown, s is a

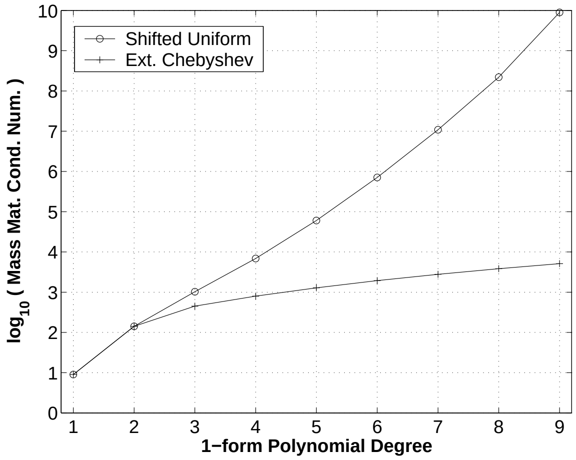


Fig. 2. Condition number of 1-form mass matrix using two different sets of interpolation points.

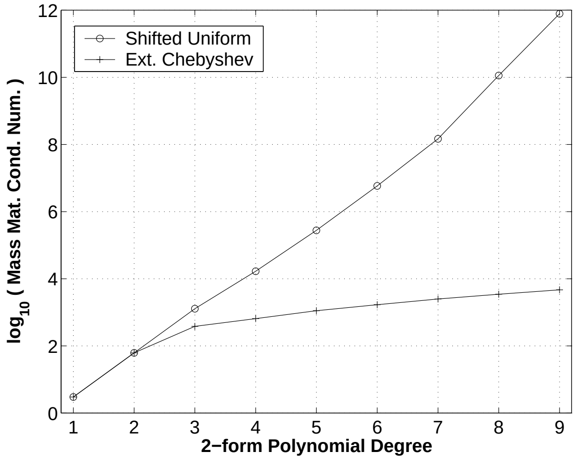


Fig. 3. Condition number of 2-form mass matrix using two different sets of interpolation points.

time harmonic source or load vector and $l = 0, 1, 2$ is the degree of the differential form. In the following examples the symbol Ω denotes the computational domain and $\partial\Omega$ denotes its surface, the symbols g, s and h denote arbitrary functions.

A. Poisson Equation

The Poisson equation corresponds to the case $l = 0$ and $\omega = 0$ from (29) and is of the form

$$\begin{aligned} \nabla \cdot (\gamma \nabla g) &= -s \text{ in } \Omega \\ g &= h \text{ on } \partial\Omega \end{aligned} \tag{30}$$

where s is a source function (e.g. electric charge). For CEM, this equation corresponds to the electrostatic field potential due to a distribution of point charges described by the function γ. In this case, the degrees of freedom from (17) will have units of voltage while the basis functions of (4) will be dimensionless. The discrete system of (29) is solved iteratively to a tolerance of 10^{-10} using a conjugate gradient algorithm with a diagonally scaled (or point Jacobi) preconditioner. Results are shown in Figure 4.

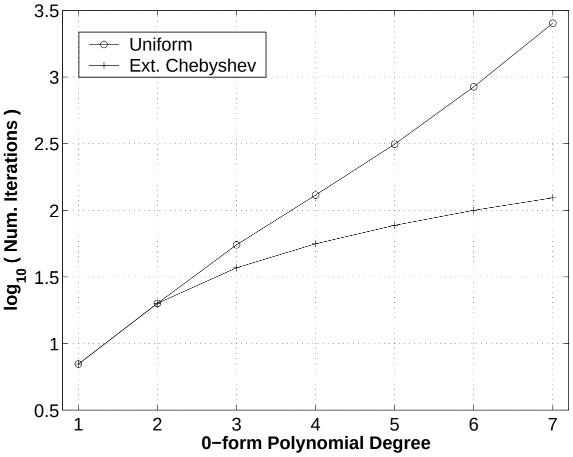


Fig. 4. Fixed mesh iteration count vs. polynomial degree for diagonally scaled PCG linear solve of discrete Poisson equation using two different types of interpolatory 0-form basis functions.

B. Vector Helmholtz Equation

The vector Helmholtz equation corresponds to the case $l = 1$ from (29) and is of the form

$$\begin{aligned} \nabla \times (\gamma \nabla \times \mathbf{g}) - \omega^2 \mathbf{g} &= \mathbf{s} \text{ in } \Omega \\ \mathbf{g} \times \hat{\mathbf{n}} &= \mathbf{h} \text{ on } \partial\Omega \end{aligned} \tag{31}$$

where $\mathbf{s}$ is a vector source function (e.g. current). For frequency domain CEM, this equation is very important; it corresponds to the second order wave equation for the electric field due to a distribution of time harmonic current (or voltage) sources; the function γ represents the inverse of the permeability of the material in which the electric field exists. It can also be used to represent the resonant modes of a cavity. For this equation, the degrees of freedom from (18) will have units of voltage while the basis functions of (9) will have units of m^{-1}. As mentioned in [8], the LHS of the discrete version of this equation is not positive definite, therefore precluding the use of a PCG algorithm. A such, we employ a generalized minimum residual (GMRES) method instead. The discrete system of (29) is solved iteratively to a residual error tolerance of 10^{-8} using an ILU preconditioned GMRES algorithm with a Krylov subspace dimension (or restart length) of 2500. Results are shown in Figure 5. Note that the large jump in iteration count at $p = 6$ for the shifted-uniform basis is most likely due to the value of the restart length.

C. Acoustic Wave Equation

The acoustic wave equation corresponds to the case $l = 2$ from (29) and is of the form

$$\begin{aligned} \nabla(\gamma \nabla \cdot \mathbf{g}) + \omega^2 \mathbf{g} &= -\mathbf{s} \text{ in } \Omega \\ \mathbf{g} \cdot \hat{\mathbf{n}} &= \mathbf{h} \text{ on } \partial\Omega \end{aligned} \tag{32}$$

where $\mathbf{s}$ is a vector source function. For CEM, the second order grad-div operator of this equation arises in the field of magnetohydrodynamics (MHD) where it is used to describe electromagnetic wave propagation in a magnetized plasma. In this case the 2-form field $\mathbf{g}$ is a pressure flux density, the degrees of freedom

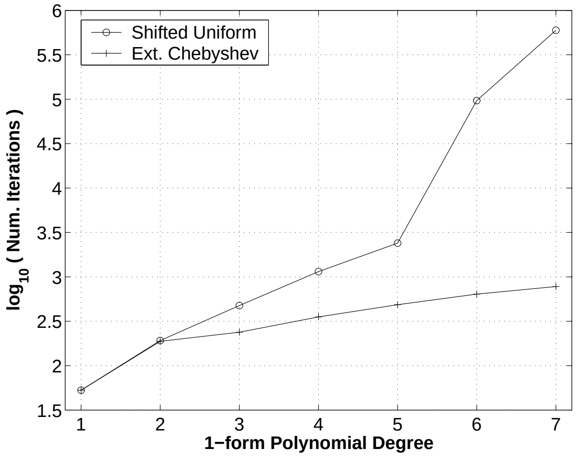


Fig. 5. Fixed mesh iteration count vs. polynomial degree for ILU preconditioned GMRES linear solve of discrete vector Helmholtz equation using two different types of interpolatory 1-form basis functions.

from (19) will have units of force while the basis functions of (13) will have units of m^{-2}. The discrete system is solved iteratively to a tolerance of 10^{-10} using a diagonally scaled preconditioned conjugate gradient algorithm. Results are shown in Figure 6.

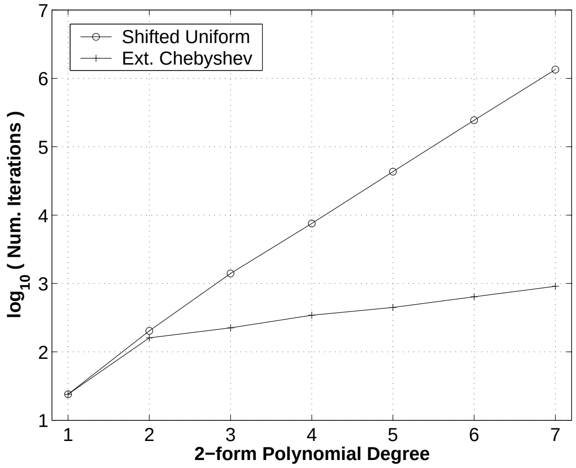


Fig. 6. Fixed mesh iteration count vs. polynomial degree for diagonally scaled PCG linear solve of discrete acoustic wave equation using two different types of interpolatory 2-form basis functions.

VII. Time Domain Examples

In these examples we solve Maxwell's equations directly in the time domain using the first order symplectic integration method of [13]. We solve the coupled first order Ampere-Faraday equations which are discretized in space using a mixed finite element formulation. This yields the following linear system of ODEs

$$\begin{aligned} M_\varepsilon \frac{\partial}{\partial t} e &= K^T M_\mu\, b - M_\varepsilon\, j \qquad (33) \\ \frac{\partial}{\partial t} b &= -K\, e \end{aligned}$$

where e and b represent the discrete differential 1-form and 2-form electric and magnetic fields respectively, M_ε is the 1-form mass matrix computed using the material property function ε to represent the dielectric properties, M_μ is the 2-form mass matrix computed using the material property function μ^{-1} to represent the magnetic permeability and j is the discrete 2-form time dependent current source. The rectangular matrix K is derived from a special mixed bilinear form

$$D_\alpha\langle \mathbf{w}_i, \mathbf{f}_j\rangle = \int_\Omega (\alpha\, \nabla\times\mathbf{w}_i)\cdot\mathbf{f}_j = M_\alpha\langle \mathbf{f}_i, \mathbf{f}_j\rangle\, K_{i,j} \qquad (34)$$

resulting in a product of the 2-form mass matrix and a new matrix K which we refer to as the *topological* derivative matrix. A topological derivative matrix for 1-forms and 2-forms is a discrete version of the curl operator and is an incidence map between the discrete differential 1-form and 2-form degrees of freedom. Specifically, the topological derivative matrix is of the form

$$K_{i,j} = \mathcal{A}_i(\nabla\times\mathbf{w}_j) \qquad (35)$$

where $\mathcal{A}_i$ are the 2-form degrees of freedom from (19). In other words, we construct this matrix by *projecting* the curl of the 1-form basis functions from (9) onto the dual space of the 2-form degrees of freedom. Stated another way, we can write the curl of a 1-form as a linear combination of the 2-form basis functions. The resulting rectangular matrix contains only topological information and is independent of the mesh geometry (since the **J** terms cancel out). It will have a number of rows equal to the dimension of the discrete 2-form basis and a number of columns equal to the dimension of the discrete 1-form basis. For the case of first order basis functions (i.e. $p = 1$), this matrix is the edge-face topological incident map commonly found in FDTD and FE methods, consisting of ±1's and 0's [14]. Equation (35) is a generalization of this notion to higher-order basis functions.

In each of the following examples, the 1-form mass matrix M_ε is solved once per time step with a residual error tolerance of 10^{-10} using a preconditioned conjugate gradient algorithm.

A. Resonant Spherical Cavity

In this example we simulate a resonant spherical cavity subject to a perfect electric conductor (PEC) boundary condition. The cavity is represented by a very coarse 32 element mesh (shown in Figure 7) with second order curved surface elements to accurately model the geometry of the sphere. We generate an oscillating electromagnetic field inside the cavity by applying a time dependent vector valued current source to a random sampling of the interior degrees of freedom. The simple current source has a temporal profile equal to the second derivative of a Gaussian pulse. This simulation can be used to compute the resonant modes of the cavity by Fourier transforming the time dependent values of the interior degrees of freedom. The problem is discretized in space using interpolatory basis functions of degree $p = 5$ resulting in a dense 1-form mass matrix of dimension $12,640$ with a total of $9,030,800$ non-zero entries. We let the simulation run for a total of 500 time steps.

We now compare the results of the linear solve performed at each time step using three different types of preconditioners.

Typically, the performance of a preconditioner is gauged by the number of iterations required to achieve some prescribed error tolerance. However, a reduction in iteration count does not always imply a reduction in total computational time; the cost of construction and application of the preconditioner must be taken into account as well.

In Table III we compare the results of the linear solve performed at each time step using the Silvester-Lagrange (SL) basis of [3] and the newly proposed extended Chebyshev (EC) basis using a point Joacobi preconditioner. This simple preconditioner requires minimal computational overhead to construct and apply, but leads to a modest reduction in iteration count. For this case, the EC basis runs roughly 11 times faster than the SL basis. In Table IV we compare results for the same problem using a sparse approximate inverse preconditioner [15]; in particular, the algorithm developed by [16]. This preconditioner requires more construction and application time than point Jacobi, but leads to a more drastic reduction in iteration count; resulting in a lower total run time. In this case, the EC basis runs about 3 times faster than the SL basis. Finally, in Table V we compare the results for the same problem using a parallel ILU preconditioner [17]. Note that for this case, application of the PILU preconditioner results in essentially the same performance for each basis, substantially reducing the number of iterations required per step in comparison to the previous preconditioners. However, because of the dense nature of the linear system, construction costs and application of the preconditioner at each time step require more total CPU time than the sparse approximate inverse preconditioner.

	SL Basis	**EC Basis**
PreCond. Setup Time	∼0.0 sec	∼0.0 sec
Avg. Iterations/step	629	51
Avg. CPU time/step	35.9 sec	3.27 sec
Total Run Time	**299.2 min**	**27.3 min**

TABLE III
COMPARISON OF RESULTS FOR RESONANT CAVITY SIMULATION WITH POINT JACOBI PRECONDITIONING

	SL Basis	**EC Basis**
PreCond. Setup Time	185.3 sec	6.03 sec
Avg. Iterations/step	66	24
Avg. CPU time/step	4.80 sec	1.71 sec
Total Run Time	**43.1 min**	**14.4 min**

TABLE IV
COMPARISON OF RESULTS FOR RESONANT CAVITY SIMULATION WITH SPARSE APPROXIMATE INVERSE PRECONDITIONING

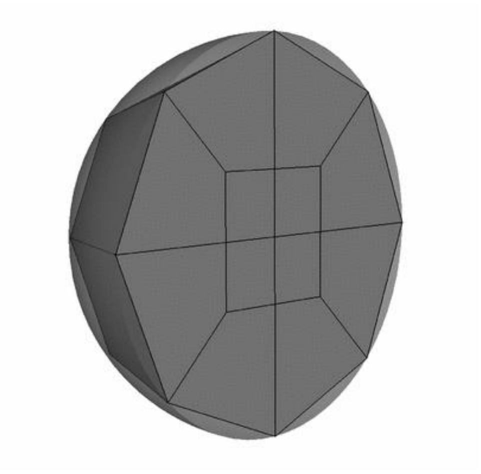


Fig. 7. Very coarse spherical cavity mesh with curvilinear surface projection elements.

B. Coaxial Waveguide

In this example we simulate the propagation of an EM wave along a coaxial waveguide. We use a very coarse 384 element mesh (shown in Figure 8) with second order curved elements to accurately model the geometry of the inner and outer cylindrical walls. The problem is excited with a time dependent voltage source boundary condition applied to the input cap of the mesh. The voltage source has a temporal profile equal to a ramped sine wave function and a spatial profile proportional to the inverse of the radial coordinate. A PEC boundary condition is applied to the inner and outer cylindrical walls while an absorbing boundary condition (ABC) is applied to the end cap of the mesh. The problem is discretized in space using interpolatory basis functions of degree $p = 4$. In Table VI we compare the results of the linear solve performed at each time step for the SL basis and EC basis using a point Jacobi preconditioner. Note that the EC basis runs at a substantially faster rate than the SL basis.

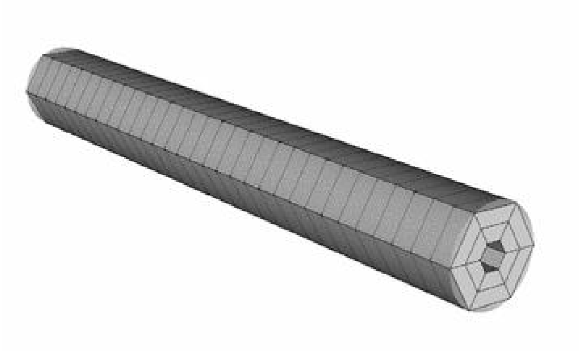


Fig. 8. Very coarse coaxial waveguide mesh with curvilinear surface projection elements.

VIII. CONCLUSIONS

We have proposed a new set of extended Chebyshev interpolatory basis functions which make use of a special set of nonuniform interpolation points. We have demonstrated the improved performance for iterative solutions to linear systems involving finite element matrices which use these basis functions in comparison to the standard Silvester-Lagrange interpolatory bases. The basis functions presented here are no more computationally expensive to implement. Preconditioners can improve the condition number of a matrix and lead to reduced iteration

	SL Basis	EC Basis
PreCond. Setup Time	38.9 min	39.3 min
Avg. Iterations/step	5	5
Avg. CPU time/step	5.87 sec	5.91 sec
Total Run Time	**87.9 min**	**88.5 min**

TABLE V

Comparison of results for resonant cavity simulation with PILU preconditioning

	SL Basis	EC Basis
Physical Time	107.5 sec	107.5 sec
Time Step	0.05 sec	0.05 sec
No. Steps	2,150	2,150
No. Unknowns	80,280	80,280
No. Nonzeros	33,055,200	33,055,200
Avg. Iterations/step	225	35
Avg. CPU time/step	12.26 sec	2.33 sec
Total Run Time	**439.3 min**	**83.5 min**

TABLE VI

Comparison of results for coaxial waveguide simulation with point Jacobi preconditioning

count. However, preconditioners are often complicated and expensive to construct and apply. For linear systems of the form $Ax = b$, a preconditioner of the form $R^T AR$ (a congruence transformation) is in fact just a change of basis; R is the matrix that maps the old basis to the new basis. In this sense, the proposed EC basis can be viewed as a preconditioner applied at the element level rather than the matrix level, resulting in better conditioned finite element matrices at no cost. In addition, we have presented the key ingredients necessary to construct these finite element matrices; namely, the specific interpolation points, the basis functions, the degrees of freedom and bilinear forms.

References

[1] H. Igarashi and T. Honma. Convergence of preconditioned conjugate gradient method applied to driven microwave problems. *IEEE Trans. Mag.*, 39(3):1705–1708, 2003.

[2] H. Igarashi and T. Honma. On convergence of ICCG applied to finite-element equation for quasi-static fields. *IEEE Trans. Mag.*, 38(2):565–568, 2002.

[3] R. Graglia, D. Wilton, and A. Peterson. Higher order interpolatory vector bases for computational electromagnetics. *IEEE Trans. Ant. Prop.*, 45(3):329–342, 1997.

[4] S. Warren and W. Scott. An investigation of numerical dispersion in the vector finite element method using quadrilateral elements. *IEEE Trans. Ant. Prop.*, 42(11):1502–1508, 1994.

[5] D. A. White. Numerical dispersion of a vector finite element method on skewed hexahedral grids. *Commun. Numer. Meth. Engng.*, 16:47–55, 2000.

[6] J. Melenk. On condition numbers in hp-fem with gauss-lobatto shape functions. *J. Comput. Appl. Math.*, 139(1):21–48, 2002.

[7] M. Ainsworth and J. Coyle. Conditioning of hierarchic p-version Nedelec elements on meshes of curvilinear quadrilaterals and hexahedra. *SIAM J. Num. Anal.*, 41(2):731–750, 2003.

[8] D. Sun, J. Lee, and Z. Cendes. Construction of nearly orthogonal Nedelec bases for rapid convergence with multilevel preconditioned solvers. *SIAM J. Sci. Comp.*, 23(4):1053–1076, 2001.

[9] G. Rodrigue, D. White, and J. Koning. Scalable preconditioned conjugate gradient inversion of vector finite element mass matrices. *J. Comput. Appl. Math.*, 123:307–321, 2000.

[10] L. Brutman. Lebesgue functions for polynomial interpolation - a survey. *Annals of Numerical Mathematics*, 4, 1997.

[11] R. Hiptmair. Canonical construction of finite elements. *Math. Comp.*, 68(228):1325–1346, 1999.

[12] J. C. Nédélec. Mixed finite elements in R3. *Numer. Math.*, 35:315–341, 1980.

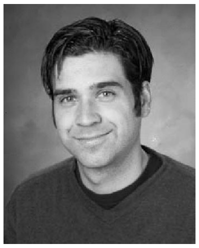
[13] R. Rieben, D. White, and G. Rodrigue. High order symplectic integration methods for finite element solutions to time dependent maxwell equations. *IEEE Trans. Ant. Prop.*, August 2004. in press.

[14] P. R. Kotiuga. Helicity functionals and metric invariance in three dimensions. *IEEE Trans. Mag.*, 25(4):2813–2815, 1989.

[15] M. Alléon, M. Benzi, and L. Giraud. Sparse approximate inverse preconditioning for dense linear systems arising in computational electromagnetics. *Numerical Algorithms*, 16(1):1–15, 1997.

[16] E. Chow. A priori sparsity patterns for parallel sparse approximate inverse preconditioners. *SIAM J. Sci. Comp.*, 21(5):1804–1822, 2000.

[17] D. Hysom and A. Pothen. A scalable parallel algorithm for incomplete factor preconditioning. *SIAM J. Sci. Comp.*, 22(6):2194–2215, 2001.



Robert N. Rieben (IEEE student member) received his B.S. in physics from the University of California at Riverside in 1999. He is presently a graduate student in the Department of Applied Science, University of California at Davis, finishing up his Ph.D. in engineering / applied science with an emphasis in computational science. His research interests include massively parallel physics simulations using high order numerical methods, time-domain vector finite element methods, the modeling and simulation of electromagnetic wave propagation and photonic crystals and photonic band-gap structures.

Daniel A. White (M'88) received the B.S. and M.S. degrees in Electrical and Computer Engineering, and the Ph.D. degree in Applied Science, from the University of California at Davis, in 1985, 1986, and 1997, respectively. From 1986 to 1993 he was a Senior Engineer with Hughes Missile Systems Company (formerly General Dynamics Convair Division) in San Diego, CA. At Hughes, Dr. White was involved with inverse synthetic aperture radar imaging, radar target identification, and low-observables technology. In 1993 he joined Lawrence Livermore National Laboratory, where he is currently Group Leader of the Computational Engineering Group of the Defense Sciences Engineering Division. His current research interests include higher-order methods for computational electromagnetics, multi-physics simulations, and massively parallel computing.

Garry H. Rodrigue received his Ph.D. degree from the University of Southern California, Los Angeles, California for work in numerical mathematics. He is currently a Professor with the Department of Applied Science, University of California at Davis, California. He is also an adjunct member of the Institute for Scientific Computing Research at the Lawrence Livermore National Laboratory in Livermore, California. His research interests are in large scale numerical simulations of phenomenon in electromagnetics, optics, fluid dynamics and magnetohydrodynamics.